

# Wikipedia Edit Number Prediction based on Temporal Dynamics Only

Dell Zhang

DCSIS

Birkbeck, University of London

Malet Street

London WC1E 7HX, UK

Email: dell.z@ieee.org

**Abstract**—In this paper, we describe our approach to the Wikipedia Participation Challenge which aims to predict the number of edits a Wikipedia editor will make in the next 5 months. The best submission from our team, “zeditor”, achieved 41.7% improvement over WMF’s baseline predictive model and the final rank of 3rd place among 96 teams. An interesting characteristic of our approach is that only temporal dynamics features (i.e., how the number of edits changes in recent periods, etc.) are used in a self-supervised learning framework, which makes it easy to be generalised to other application domains.

**Keywords**—social media; user modelling, data mining; machine learning.

## I. INTRODUCTION

Wikipedia is “a free, web-based, collaborative, multi-lingual encyclopaedia project” supported by the non-profit Wikimedia Foundation (WMF). Started in 2001, Wikipedia has become the largest and most popular general reference knowledge source on the Internet. Almost all of its 19.7 million articles can be edited by anyone with access to the site, and it has about 90,000 regularly active volunteer editors around the world. However, it has recently been observed that Wikipedia growth has slowed down significantly [1]. In particular, WMF has reported that<sup>1</sup>:

Between 2005 and 2007, newbies started having real trouble successfully joining the Wikimedia community. Before 2005 in the English Wikipedia, nearly 40% of new editors would still be active a year after their first edit. After 2007, only about 12-15% of new editors were still active a year after their first edit. Post-2007, lots of people were still trying to become Wikipedia editors. What had changed, though, is that they were increasingly failing to integrate into the Wikipedia community, and failing increasingly quickly. The Wikimedia community had become too hard to penetrate.

It is therefore of utter importance to understand quantitatively what factors determine editors’ future editing behaviour (why they continue editing, change the pace of editing, or stop editing), in order to ensure that the Wikipedia

community can continue to grow in terms of size and diversity.

The Wikipedia Participation Challenge<sup>2</sup>, sponsored by WMF and hosted by Kaggle, request contestants to build a predictive model that could accurately predict the number of edits a Wikipedia editor would make in the next 5 months based on his edit history so far. Such a predictive model may be able to help WMF in figuring out how people can be encouraged to become, and remain, active contributors to Wikipedia.

The ‘training’ dataset consists of randomly sampled active editors with their full history of editing activities on the English Wikipedia (the first 6 namespaces only) in the period from 2001-01-01 to 2010-09-01. An editor is considered “active” if he or she made at least one edit in the last one year period, i.e., from 2009-09-01 to 2010-09-01. For each edit, the available information includes its user\_id, article\_id, revision\_id, namespace, timestamp, etc.

The predictive model to be constructed should predict, for each editor from the ‘training’ dataset, how many edits would be made in the 5 months after the end date of the ‘training’ dataset, i.e., from 2010-09-01 to 2011-02-01. The predictive model’s accuracy is going to be measured by the Root Mean Squared Logarithmic Error (RMSLE):

$$\epsilon = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(1 + p_i) - \log(1 + a_i))^2}, \quad (1)$$

where  $n$  is total number of editors in the dataset,  $\log(\cdot)$  is the natural logarithm function,  $p_i$  and  $a_i$  are the predicted and actual edit numbers respectively for editor  $i$  in the next 5 month period.

The best submission from our team, “zeditor”, achieved 41.7% improvement over WMF’s baseline predictive model and the final rank of 3rd place among 96 teams. An interesting characteristic of our approach is that only temporal dynamics features (i.e., how the number of edits changes in recent periods, etc.) are used in a self-supervised learning framework, which makes it easy to be generalised to other application domains.

<sup>1</sup>[http://strategy.wikimedia.org/wiki/March\\_2011\\_Update](http://strategy.wikimedia.org/wiki/March_2011_Update)

<sup>2</sup><http://www.kaggle.com/c/wikichallenge>

The rest of this paper is organised as follows. In Section II, we present our approach in details. In Section III, we show the experimental results. In Section IV, we review the related work. In Section V, we make conclusions.

## II. APPROACH

Our basic idea is to build a predictive model  $f$  (that estimates an active editor’s future number of edits based on his recent edit history) through self-supervised learning, as illustrated schematically in Figure 1. The approach is called “self-supervised” to emphasise the fact that it does not require any manual labelling of data (as in standard *supervised learning* [2]) but extracts the needed labels from data automatically.

To facilitate the description of our approach, we shall from now on talk about any time-length in the unit of months and refer to any time-point as the real number of months passed since the beginning date of the dataset. So for the official dataset ‘training’, the timestamp “2001-06-16 00:00:00” would be 5.5 because it is five and a half months since 2001-01-01.

Let  $t_{\text{test}}$  denote the time-point when we would like to predict each active editor’s number of edits in the next 5 months. To train the predictive model, we would move 5 months backwards and assume that we were at the time-point  $t_{\text{train}} = t_{\text{test}} - 5$ . Thus we could know the actual number of edits made by each active editor in those 5 months after  $t_{\text{train}}$ , i.e., the label for our machine learning (regression) methods. Specifically, the target value for regression would be set as  $y_i = \log(1 + a_i)$  where  $a_i$  is the actual number of edits in the next 5 months. In this way, the *squared error* loss function  $L(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$  used by most machine learning methods (including those in our experiments and final submission) would connect the *empirical risk* [2] directly to the evaluation metric RMSLE:

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^n L(f(\mathbf{x}_i), y_i) = \epsilon^2. \quad (2)$$

Given a time-point (either  $t_{\text{train}}$  or  $t_{\text{test}}$ ), each active editor  $i$  would be represented as a vector  $\mathbf{x}_i$  that consists of the following temporal dynamics features:

- the number of edits in recent periods of time;
- the number of edited articles in recent periods of time;
- the length of time between the first edit and the last edit, scaled logarithmically.

The periods used in our final submission for the above temporal dynamics features are

$$\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 12, 36, 108$$

where the length of period first doubles at each step from  $\frac{1}{16}$  to 4 and then triples at each step from 4 to 108. The usage of such temporal dynamics features was inspired by the decent performance of the “most-recent-5-months-benchmark” —

Table I  
THE CHARACTERISTICS OF EACH DATASET.

dataset	$t_{\text{train}}$	$t_{\text{test}}$	#editors	#edits
‘validation’	79	84	4856	274820
‘moredata’	106	111	23584	5717049
‘training’	111	116	44514	22326031

if using the exact number of edits in just one period (the last 5 months) for prediction could work reasonably well, we should be able to achieve a better performance by using many more recent periods. The periods were chosen to be at exponentially increasing temporal scales, because we conjecture that the influence of an editing activity to the editor’s future editing behaviour would be exponentially decaying along with the time distance away from now. The process of *exponential decay*<sup>3</sup> occurs in numerous natural phenomena, and it has been widely used in temporal applications where it is desirable to gradually discount the history of past events [3]. One reason for changing from doubling to tripling midway through is to include the special period of 12-months (i.e., one year) that has been used to define the “active” editors. The periods will be capped by the time scope of the given dataset (e.g., 106 for the additional dataset ‘moredata’) in case they are out of range.

We have also introduced a constant *drift* term (i.e., how much the average number of edits would change after 5 months) into the formula of making final predictions, which is a crude way to cover the global shift of target values along with time. Again, its value is estimated from the situation 5 months ago.

The concise pseudo-code of our algorithms for learning and predicting is shown in Figure 2. The complete source code will be made available at the author’s homepage<sup>4</sup>.

## III. EXPERIMENTS

### A. Datasets

There are three datasets available to all contestants:

- ‘training’ is the official dataset for training and testing;
- ‘validation’ is the official dataset for validation;
- ‘moredata’ is the additional dataset generously provided by Twan van Laarhoven<sup>5</sup>.

The characteristics of each dataset are shown in Table I.

Since we did not have local access to the true labels (target values) of the dataset ‘training’, we only used it to make the final submission, but conducted our experiments (for parameter tuning etc.) on the other two datasets ‘validation’ and ‘moredata’. It is noteworthy that these two datasets ‘validation’ and ‘moredata’ had been filtered to contain only

<sup>3</sup>[http://en.wikipedia.org/wiki/Exponential\\_decay](http://en.wikipedia.org/wiki/Exponential_decay)

<sup>4</sup><http://www.dcs.bbk.ac.uk/~dell/>

<sup>5</sup><http://www.kaggle.com/c/wikichallenge/forums/t/719/more-training-data>

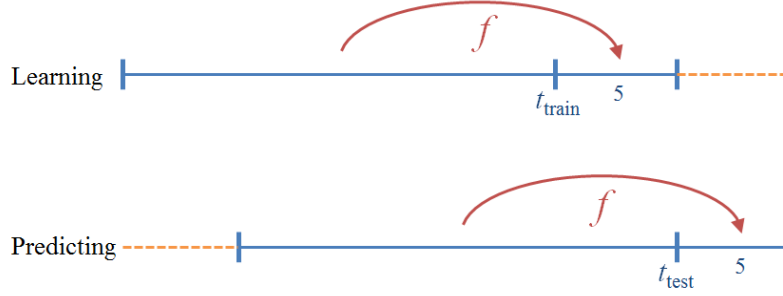


Figure 1. Our self-supervised learning framework.

---

### Learning

- $t = t_{\text{train}}$  (i.e.,  $t_{\text{test}} - 5$ )
  - for each active editor  $i$  who made at least one edit in  $[t - 12, t)$ :
    - represent the editor as a vector  $\mathbf{x}_i$  consisting of temporal dynamics features (please refer to the above description)
    - label the editor by  $y_i = \log(1 + a_i)$  where  $a_i$  is the actual number of edits in  $[t, t + 5)$
  - learn a predictive model/function  $f : x \rightarrow y$  from  $(\mathbf{x}_i, y_i)$  pairs using a regression technique such as GBT
  - estimate the drift  $d$  by comparing the average number of edits in  $[t - 5, t)$  and that in  $[t, t + 5)$
- 

### Predicting

- $t = t_{\text{test}}$  (e.g., 116 for the dataset ‘training’)
  - for each active editor  $i$  who made at least one edit in  $[t - 12, t)$ :
    - represent the editor as a vector  $\mathbf{x}_i$  consisting of temporal dynamics features (please refer to the above description)
    - compute  $\hat{y}_i = f(\mathbf{x}_i)$  using the learnt  $f$
    - output  $p_i = \exp(\max(\hat{y}_i + d, 0)) - 1$  as the predicted number of edits in  $[t, t + 5)$
- 

Figure 2. Our algorithms for learning and predicting.

active editors (who made at least one edit in the last one year period) in order to make them exhibit the same *survivorship bias*<sup>6</sup> as the dataset ‘training’. This might (partially) ensure that the experimental findings on the former two datasets could be transferred to the latter one.

### B. Tools

We have only used Python<sup>7</sup> (equipped with Numpy<sup>8</sup>) to write small programs for analysing data and making predictions. The machine learning methods that we have tried for our regression task all come from two *open-source* Python modules: one is scikit-learn<sup>9</sup>, and the other is OpenCV<sup>10</sup>.

### C. Results

First, we compare different machine learning methods (with their default parameter values) in terms of their predic-

tion performances (RMSLE). The methods being compared include:

- Ordinary Least Squares (OLS)<sup>11</sup>,
- Support Vector Machine (SVM)<sup>12</sup>,
- K Nearest Neighbours (KNN)<sup>13</sup>,
- Artificial Neural Network (ANN)<sup>14</sup>,
- Gradient Boosted Trees (GBT)<sup>15</sup>.

The experimental results are shown in Table II and Figure 3. Gradient Boosted Trees (GBT)<sup>16</sup> [4], [5] clearly outperformed all the other machine learning methods on both datasets. GBT (aka GBM, MART and TreeNet) represents a general and powerful machine learning method that builds an ensemble of *weak* tree learners in a greedy fashion. It evolved from the application of boosting to regression trees [2]. The general idea is to compute a sequence of very simple

<sup>6</sup>[http://en.wikipedia.org/wiki/Survivorship\\_bias](http://en.wikipedia.org/wiki/Survivorship_bias)

<sup>7</sup><http://www.python.org/>

<sup>8</sup><http://numpy.scipy.org/>

<sup>9</sup><http://scikit-learn.sourceforge.net/>

<sup>10</sup><http://opencv.willowgarage.com/wiki/>

<sup>11</sup>[http://scikit-learn.sourceforge.net/modules/linear\\_model.html#ordinary-least-squares-ols](http://scikit-learn.sourceforge.net/modules/linear_model.html#ordinary-least-squares-ols)

<sup>12</sup><http://scikit-learn.sourceforge.net/modules/svm.html>

<sup>13</sup>[http://opencv.itseez.com/modules/ml/doc/k\\_nearest\\_neighbors.html](http://opencv.itseez.com/modules/ml/doc/k_nearest_neighbors.html)

<sup>14</sup>[http://opencv.itseez.com/modules/ml/doc/neural\\_networks.html](http://opencv.itseez.com/modules/ml/doc/neural_networks.html)

<sup>15</sup>[http://opencv.itseez.com/modules/ml/doc/gradient\\_boosted\\_trees.html](http://opencv.itseez.com/modules/ml/doc/gradient_boosted_trees.html)

<sup>16</sup>[http://en.wikipedia.org/wiki/Gradient\\_boosting](http://en.wikipedia.org/wiki/Gradient_boosting)

Table II  
THE PREDICTION PERFORMANCES OF DIFFERENT MACHINE LEARNING METHODS (WITH THEIR DEFAULT PARAMETER VALUES).

learning method	‘validation’	‘moredata’
OLS	0.832351	0.869779
SVM	0.901698	0.732814
KNN	0.833288	0.690832
ANN	0.987345	1.040396
GBT	0.820805	0.635807

Table III  
THE PREDICTION PERFORMANCES OF GBT WITH DIFFERENT NUMBER OF WEAK TREE LEARNERS (WEAK\_COUNT).

GBT <i>weak_count</i>	‘validation’	‘moredata’
200	0.820805	0.635807
400	0.817789	0.616876
600	0.817483	0.614507
800	0.817614	0.613757
1000	0.818726	0.613530
1200	0.819804	0.613465
1400	0.819998	0.613671

trees, where each successive tree is built for the prediction residuals of all preceding trees on a randomly selected sub-sample of the full training dataset. Eventually a “weighted additive expansion” of those trees can produce an excellent fit of the predicted values to the observed values. It allows optimisation of any differentiable loss function. Here we just use the *squared error* for the reasons given in Section II. The success of GBT in our task is probably attributable to (i) its ability to capture the complex nonlinear relationship between the target variable and the features, (ii) its insensitivity to different feature value ranges as well as outliers, and (iii) its resistance to overfitting via regularisation mechanisms such as shrinkage and subsampling [4], [5].

Second, we investigate how GBT’s most important parameter *weak\_count* — the number of weak tree learners — affects its prediction performance for our task. Tuning *weak\_count* is our major means of controlling the model complexity to avoid underfitting or overfitting. The experimental results are shown in Table III and Figure 4. It seems that on big datasets like ‘moredata’, a higher value of *weak\_count* (i.e., more weak tree learners) would be beneficial, but on small datasets like ‘validation’, it might increase the risk of overfitting.

Third, we demonstrate how the prediction performance changes when we use more and more periods to generate temporal dynamics features: we start from just the shortest period ( $\frac{1}{16}$ ) and then each time we add the next longer period to the series (see Section II). The experimental results are shown in Table IV and Figure 5. It seems that making use of more periods for temporal dynamics features usually helps, but the pay-off gradually diminishes.

Table IV  
THE PREDICTION PERFORMANCES OF GBT USING DIFFERENT NUMBER OF PERIODS FOR TEMPORAL DYNAMICS FEATURES.

GBT #periods	‘validation’	‘moredata’
1	0.861111	0.788450
2	0.857575	0.760365
3	0.849440	0.728888
4	0.841127	0.696196
5	0.836116	0.669754
6	0.830619	0.647883
7	0.829393	0.629062
8	0.816459	0.614429
9	0.818515	0.613749
10	0.818726	0.613530

#### D. Submissions

Since ‘moredata’ is more similar than ‘validation’ to the official dataset ‘training’ in terms of the time scope and the number of editors, we applied the best working algorithm, GBT, with the optimal parameter setting on ‘moredata’ (*weak\_count* = 1000), to make the final submission based on ‘training’. It got an RMSLE score of 0.862582 on the private leaderboard, which is roughly 41.7% better than WMF’s baseline predictive model. The final rank of our team, “zeditor”, is the 3rd place among 96 teams.

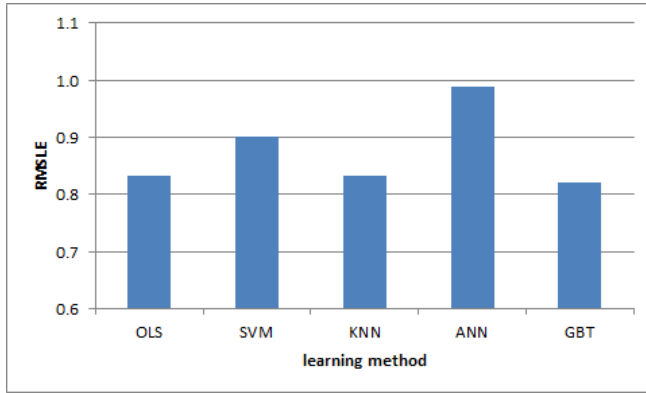
#### IV. RELATED WORK

The global slowdown of Wikipedia’s growth rate (both in the number of editors and the number of edits per month) has been studied [1]. It is found that medium-frequency editors now cover a lower percentage of the total population while high frequency editors continue to increase the number of their edits. Moreover, there are increased patterns of conflict and dominance (e.g., greater resistance to new edits in particular those from occasional editors), which may be the consequence of the increasingly limited opportunities in making novel contributions. These findings could guide us to generate other kinds of useful features to tackle the problem of edit number prediction. Furthermore, researchers have also investigated other activities of Wikipedia’s editors, such as voting on the promotion of Wikipedia admins [6].

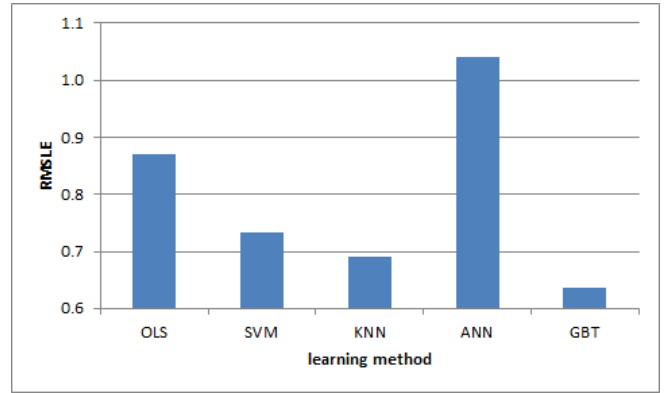
In addition to Wikipedia, the temporal dynamics of online users’ behaviour has been explored and exploited in web search [7]–[10], social tagging [11], [12], blogging [13], twittering [14], and collaborative filtering [15]. The *power law* [16] and the *exponential decay* [3] seem to be recurrent themes across application domains.

#### V. CONCLUSIONS

Our most important insight is that a Wikipedia editor’s future behaviour can be largely determined by the temporal dynamics of his recent behaviour. We are a bit surprised that just temporal dynamics features can go such a long

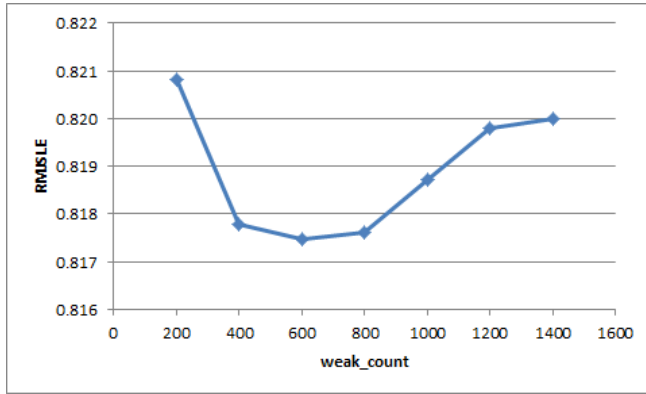


(a) validation

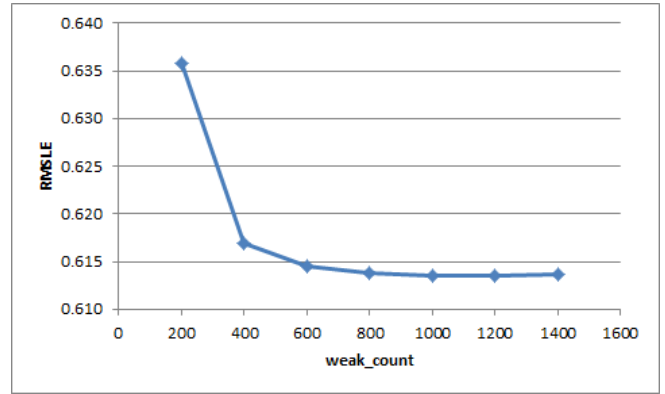


(b) moredata

Figure 3. The prediction performances of different machine learning methods (with their default parameter values).

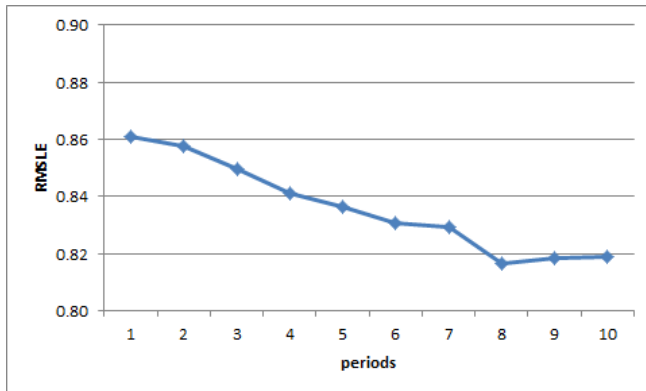


(a) validation

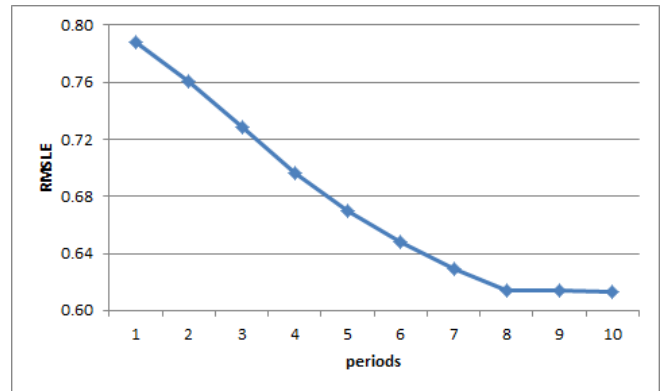


(b) moredata

Figure 4. The prediction performances of GBT with different number of weak tree learners (weak\_count).



(a) validation



(b) moredata

Figure 5. The prediction performances of GBT using different number of periods for temporal dynamics features.

way when we choose proper temporal scales and employ a powerful machine learning method. Human beings seem to be working and living in a more mechanical way than one might have thought. Since such temporal dynamics features

are actually independent of any semantics or knowledge about this specific problem, our approach could be easily generalised to other application domains, such as predicting the future supermarket spendings of shoppers (e.g., the

dunnhumby’s Shopper Challenge<sup>17</sup>), predicting the future hospital admissions of patients (e.g., the Heritage Health Prize Competition<sup>18</sup>), and so on, based on historical behavioural data.

Have we answered the question that we asked at the beginning of this paper? Yes and No. On one hand, we have built a predictive model which can be used to identify those editors who are likely to become inactive, or in other words, who need special care and attention to be kept — if an editor is going to leave the Wikipedia community, there would probably be early signals in the temporal dynamics of his recent behaviour. On the other hand, that predictive model is pretty much a black box — it does not reveal the underlying reasons why editors become inactive, and therefore it cannot tell us how to encourage editors to remain active. For the ultimate purpose of Wikipedia’s sustainable growth, we will need to investigate which attributes of an editor (his articles’ category distribution, his relationship with other editors, etc.) and also which recent events happened to him (his articles being deleted, his revisions being reverted, unfair comments about his edits being received, etc.) could affect his behaviour. Due to the time constraints and the dataset limitations (for example, the lack of information about articles and comments in the datasets ‘validation’ and ‘moredata’), we have to leave it to future work.

Long live Wikipedia!

#### ACKNOWLEDGMENT

We would like to thank WMF and Kaggle for their wonderful job in organising this interesting contest. We are grateful to Twan van Laarhoven for creating and sharing the additional dataset ‘moredata’. We also appreciate the reviewers’ helpful comments.

#### REFERENCES

- [1] B. Suh, G. Convertino, E. H. Chi, and P. Pirolli, “The singularity is not near: Slowing growth of Wikipedia,” in *Proceedings of the 2009 International Symposium on Wikis (WikiSym)*, Orlando, FL, USA, 2009.
- [2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.
- [3] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, “A framework for projected clustering of high dimensional data streams,” in *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB)*, Toronto, Canada, 2004, pp. 852–863.
- [4] J. Friedman, “Greedy function approximation: A gradient boosting machine,” IMS 1999 Reitz Lecture, Tech. Rep., Feb 1999.
- [5] —, “Stochastic gradient boosting,” Stanford University, Tech. Rep., Mar 1999.
- [6] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg, “Governance in social media: A case study of the wikipedia promotion process,” in *Proceedings of the 4th International Conference on Weblogs and Social Media (ICWSM)*, Washington, DC, USA, 2010.
- [7] D. Zhang, J. Lu, R. Mao, and J.-Y. Nie, “Time-sensitive language modelling for online term recurrence prediction,” in *Proceedings of the 2nd International Conference on the Theory of Information Retrieval (ICTIR)*, Cambridge, UK, 2009, pp. 128–138.
- [8] D. Zhang and J. Lu, “What queries are likely to recur in web search?” in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Boston, MA, USA, 2009.
- [9] J. L. Elsas and S. T. Dumais, “Leveraging temporal dynamics of document content in relevance ranking,” in *Proceedings of the 3rd International Conference on Web Search and Web Data Mining (WSDM)*, New York, NY, USA, 2010, pp. 1–10.
- [10] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais, “Understanding temporal query dynamics,” in *Proceedings of the 4th International Conference on Web Search and Web Data Mining (WSDM)*, Hong Kong, China, 2011, pp. 167–176.
- [11] D. Zhang, R. Mao, and W. Li, “The recurrence dynamics of social tagging,” in *Proceedings of the 18th International Conference on World Wide Web (WWW)*, Madrid, Spain, 2009, pp. 1205–1206.
- [12] H. Halpin, V. Robu, and H. Shepherd, “The complex dynamics of collaborative tagging,” in *Proceedings of the 16th International Conference on World Wide Web (WWW)*, Banff, Alberta, Canada, 2007, pp. 211–220.
- [13] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng, “Detecting splogs via temporal dynamics using self-similarity analysis,” *ACM Transactions on the Web (TWEB)*, vol. 2, no. 1, pp. 1–35, 2008.
- [14] F. Abel, Q. Gao, G.-J. Houben, and K. Tao, “Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web,” in *Proceedings of the 3rd International Conference on Web Science (WebSci)*, Koblenz, Germany, 2011.
- [15] Y. Koren, “Collaborative filtering with temporal dynamics,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Paris, France, 2009, pp. 447–456.
- [16] A. Clauset, C. R. Shalizi, and M. E. J. Newman, “Power-law distributions in empirical data,” *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

<sup>17</sup><http://www.kaggle.com/c/dunnhumbychallenge>

<sup>18</sup><http://www.heritagehealthprize.com/c/hhp>